International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Extreme Learning Machine with Multi-Agent System for Regression

Chong Tak Yaw, Keem Siah Yap, Shen Yuong Wong, Chin Hooi Tan

Abstract: From the point of learning speed as well as generalization, Extreme Learning Machine(ELM) is widely known as an effective learning algorithm than the conventional learning methods. Basically, hidden neurons are not required in neuron alike, instead, weight is the parameter that would need to learn about the link in between output and hidden layers. The creation of an output is to integrate each independent of several ELMs. The precise approach is included in a Multi-Agent System. The novelty of ELM-MAS (extreme learning machine based multi-agent system) is put forward in the paper for solving data regression problems. The ELMs consist of two layers which are the parent agent layer and individual agent layer. The effectiveness of the ELM-MAS model is proved with some activation functions employing benchmark datasets (abalone, strike and space-ga) and real world application (No_x emission). The outcomes indicate that the proposed model is capable to attain improved results than other approaches.

Keywords: Extreme Learning Machine (ELM); Multi Agent System (MAS); Data Regression; NO_x Emission of Power Plant

I.INTRODUCTION

Nowadays, ELM(known as Extreme Learning Machine) is well recognized as an effective learning algorithm from the point of generalization and learning speed as compared to the conventional learning approaches[1-7].In addition, the universal approximation with input weights and haphazard biases is capable for ELM [8].

ELM (based Huang et al. [9, 10]) is exceptionally incline and effective to global optimum in the comparison to the conventional FNN (feedforward neural network). Furthermore, the greatest generalization bound of the conventional FNN is able to accomplish by ELM, in which every parameter in the state of normalization is learned with activation functions [11]. Based on the diverse types of problems [1-7], ELM performs much better than that of traditional FNN. There are a lot of fields that ELM have been applied in, for example biomedical analysis [12, 13], hyperspectral images [14], power systems [15, 16], action recognition [17], chemical process [18], system modelling[19, 20], and others.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Chong Tak Yaw*, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.
Keem Siah Yap, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.
Shen Yuong Wong, Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Malaysia
Chin Hooi Tan, Business Innovation Incubator, Tenaga Nasional Berhad, Malaysia

There is some researches integration with every independent estimate of several ELMs to produce an output [21-25]. The kind of particular approach is the same as MAS (known as Multi Agent System)[26]. There are some researches that successfully applied for tackling problems in different domains, for example in health care [27-29], eCommerce[26, 30-32], military support [33, 34], knowledge management [35-38], decision support [39-42], as well as control systems[43-46]. The Fig.1 is shown that the MAS's general structure, in which the base platform is the collection of ELMs known as individual agents. Generally, an individual ELM (i.e.individual agent,)'s outcome is delivered to a place called parent agent, which is the concluding combined module.

In general, the mean output in the concluding combined module is derived from the approaches including weighted mean[7], voting [47], confusion matrix [48] and exact mean [22]. Regrettably, in the concluding combined module, these aforementioned approaches often need additional algorithms to generate outcome. In this paper, concluding combined module based on ELM is proposed.

A common method called meta-learning is used to collect and combine the results of several learners [49, 50]. This can roughly be defined as knowledge material by at least a learner [49, 50]. The model is achieved by multiple ELMs as hidden neurons to estimate outputs which are learned by the meta-learner. Theoretical investigation and experimental outcomes from several studies using the datasets of benchmark and artificial regression show that the Meta-ELM, which is trained by multiple ELMs, could give good performance at the expense of a lower computational cost[51]. Nonetheless, in this paper, an extreme learning machine-based multi agent system (ELM-MAS) is designed from alternative perception. For this proposed model, it has two layers of full ELMs:the first layer consists of a single ELM and it is the parent agent; the second layer consisted of at least an ELMs where each ELM is considered as an individual agent. This proposed ELM-MAS model with two layers' structure resembles a typical MAS(Figure 1).

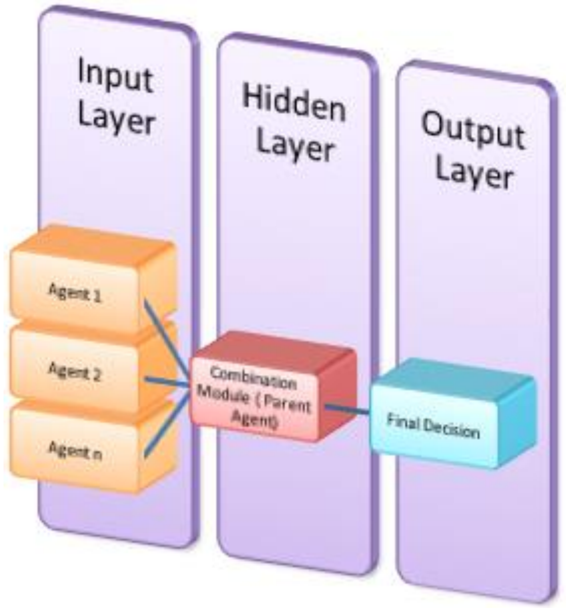

Fig. 1 A General Structure of Multi Agent System (MAS)

II.THE ALGORITHMS OF ELM AND ELM-MAS

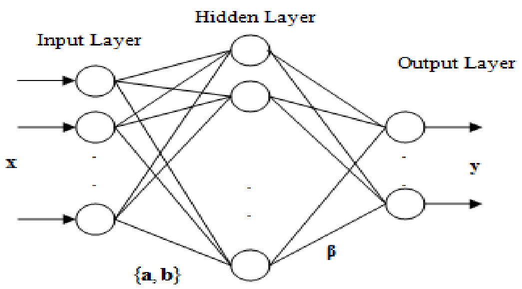

Fig. 2 The Construct of ELM

An ELM can either be a RBF or a feedforward network with a sophisticated learning algorithm because it's depending on the type of activation function it utilized as shown in Fig. 2. For example, $(\mathbf{x}_j, \mathbf{t}_j)$, i.e. $\mathbf{x}_j \in \mathbf{R}^M$ (M is quantity of input attributes), $\mathbf{t}_j \in \mathbf{R}^C$ (C is quantity of classes) and a series of N training samples (with a input vector as well as individual target output vector), is utilized to train an ELM consisting of L number of hidden neurons. As for the model in this paper, five ELMs are generated as individual agents, each had different random input weights and the class is set to 1. As shown in Figure3, the output of each ELM^k (for k = 1, 2, ...,5), in response to $\mathbf{x}_j$ is

$$\mathrm{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \mathbf{t}_j \quad (1)$$

for j = 1, …, N and for c = 1, …, C

Where $\boldsymbol{\beta}_i^k$ is the output weights, $\mathbf{a}_i^k$ and b_i^k are the bias and input weights (learning constraints) of the hidden neurons, whereas $G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$ is the output of the i^{th} hidden neuron given the input vector $\mathbf{x}_j$.

The following equations respectively reveal the description of the $G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$ for additive sigmoid hidden neuron as well as RBF hidden neuron.

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \frac{1}{1+\exp\{-(\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k)\}}, \quad (2)$$

$$b_i^k \in R$$

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-b_i^k \left\|\mathbf{x}_j - \mathbf{a}_i^k\right\|^2\}, \quad (3)$$

$$b_i^k \in R^+$$

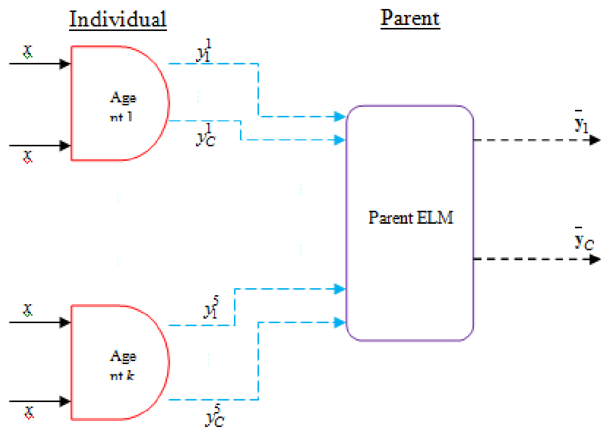

Fig. 3 Several Individual Agents and a Parent Agent for Architecture of ELM-MAS

III.THE METHODOLOGY OF ELM-MAS

The procedure of the training phase is detailed below.

Stage 1: Randomly allocate the input weights $\mathbf{a}_i^k$ and b_i^k for k = 1, 2, ..., 5and i= 1,…, L.

Stage 2: The hidden layer output matrix for ELM^k (for k = 1, 2, ...5), $\mathbf{H}^k$, is calculated as follows.

$$\mathbf{H}^k = \begin{bmatrix} G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_1) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_N) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_N) \end{bmatrix}_{N\times L} \quad (4)$$

Stage 3: Calculation of the $\boldsymbol{\beta}^k$ (output weights of ELM^k).Because of that $\mathbf{H}$ is possibly a non-symmetrical matrix, the inverse matrix cannot be solved. So to tackle this problem, the Moore-Penrose pseudo inverse matrix method is adopted as following equation,

$$\boldsymbol{\beta}^k = \left((\mathbf{H}^k)^T(\mathbf{H}^k)\right)^{-1}(\mathbf{H}^k)^T\mathbf{T}, \quad (5)$$

where $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_N]^T$ is corresponding targeted output vectors.

Stage 4: After output weights of ELM^k are calculated, the outputs of ELM^k are computed using the training samples.

$$\mathbf{y}^k = \mathrm{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$$

for j = 1, …, N and for c = 1, …, C**(6)**

Stage 5:Randomly allocate the $\mathbf{p}_i$ and q_i (input weights for parent ELM), for$i$= 1,…,L_1, where L_1 is referring to the quantity of hidden neurons of parent ELM.

Stage 6:Computation of **S**(hidden layer output matrix for parent ELM), as follow

$$\mathbf{S}=\begin{bmatrix} G(\mathbf{p}_1,q_1,\mathbf{w}_1) & \ldots & G(\mathbf{p}_{L1},q_{L_1},\mathbf{w}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{p}_1,q_1,\mathbf{w}_N) & \ldots & G(\mathbf{p}_{L_1},q_{L_1},\mathbf{w}_N) \end{bmatrix}_{N\times L_1} \quad (7)$$

Where $\mathbf{w}_j$ is the summation outputs of ELM^k (for k = 1, 2, ...5) in response to $\mathbf{x}_j$, i.e, $\mathbf{w}_j=[\mathbf{y}_j^1 \quad \mathbf{y}_j^2 \quad \mathbf{y}_j^3 \quad \mathbf{y}_j^4 \quad \mathbf{y}_j^5]$, $\mathbf{y}_j^k\in\mathbf{R}^C$, and $\mathbf{w}_j\in\mathbf{R}^{5C}$.

Stage 7:Determine the output weights of parent ELM, $\boldsymbol{\alpha}$ by the equation below,

$$\boldsymbol{\alpha}=\left((\mathbf{S})^T(\mathbf{S})\right)^{-1}(\mathbf{S})^T\mathbf{T}, \quad (8)$$

where $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_N]^T$ is the corresponding targeted output vectors.

As soon as every sample data is trained with Stage 1-7, the ELM-MAS is utilized for validation of an new input vector **z** established on the $\mathbf{a}^k$, $\mathbf{b}^k$, $\mathbf{p}$, $\mathbf{q}$, $\boldsymbol{\beta}^k$ and $\boldsymbol{\alpha}$ i.e.,

$$\mathbf{h}^k=[G(\mathbf{a}_1^k,b_1^k,\mathbf{z}) \quad \ldots \quad G(\mathbf{a}_L^k,b_L^k,\mathbf{z})]_{1\times L} \quad (9)$$

$$\mathbf{y}^k=\mathbf{h}^k\boldsymbol{\beta}^k \quad (10)$$

$$\mathbf{s}=[G(\mathbf{p}_1,q_1,\mathbf{v}) \quad \ldots \quad G(\mathbf{p}_L,q_L,\mathbf{v})]_{1\times L1} \quad (11)$$

$$\bar{\mathbf{y}}=\mathbf{s}\boldsymbol{\alpha} \quad (12)$$

Where $\mathbf{h}^k$ and $\mathbf{y}^k$ are hidden layer and output of ELM^k respectively, $\mathbf{v}=[\mathbf{y}^1 \quad \mathbf{y}^2 \quad \mathbf{y}^3 \quad \mathbf{y}^4 \quad \mathbf{y}^5]$ is combination outputs of the ELM^k in response to **z**, whereas **s** and $\bar{\mathbf{y}}$ are hidden layers of final output of the validation respectively.

After computing the output of ELM for testing samples, the RMSE (root mean squared error)is computed.

IV.EXPERIMENTAL AND RESULTS USING BENCHMARK DATA

There are three benchmark datasets (namely Abalone, Strike and Space-ga) from the UCI machine source which are used to test the capability of ELM-MAS. The specifications of datasets are detailed in Table 1 as shown below. All experimentations are run in MATLAB (version 2010) on a private laptop armed with 8G RAM and Intel(R) Core(TM) i7 2.9 GHz CPU.

Table. 1 Prescription of Benchmark Datasets

Datasets	No. of Attributes	No. of Training Samples	No. of Testing Samples	No. of Total Samples
Abalone	8	3000	1177	4177
Strike	6	416	209	625
Space-ga	6	2071	1036	3107

For the setting of the experiment, the quantity of hidden neurons of each ELM^k is fixed (i.e., L) to 180 for the three benchmark datasets. On the other hand, two-thirds of the training samples are used for training while the remaining one-third are utilized to work out the most suitable number of neurons of the parent ELM (i.e., L_1) via a validation process. The validation and training processes are started by setting L_1 = 50 units and then increased by an increment of 50 units for each type of the activation function of ELM-MAS,.

Table 2 outlines the summary of outcomes by means of ELM-MAS for all kinds of activation functions in the context of the RMSE and the quantity of hidden neurons. The best results (i.e. Laplace Basis for both abalone and strike, and Sigmoid for Space-ga) are presented in Table 2.

Table. 2 Summary of RMSE of the ELM-MAS with Different Activation Functions

Activation Functions	Abalone		Strike		Space-ga	
	RMSE	# Hidden Neurons	RMSE	# Hidden Neurons	RMSE	# Hidden Neurons
RBF	0.016639	20	0.037227	11	0.007228	12
Sigmoid	0.015363	4	0.124423	7	**0.006677**	**5**
Gaussian	0.015751	4	0.048052	10	0.007493	10
LaplaceAct	0.015975	9	0.041143	10	0.013980	7
Laplace Basis	**0.014845**	**11**	**0.018766**	**9**	0.008482	13

The outcomes of the proposed ELM-MAS are also compared to the results of other ELM-based methods. As observed from Table 3, smaller value of RMSE represents better results. Therefore, the RMSE of ELM-MAS are better as compared to SVM[51] and ELM[3].

Table. 3 RMSE of ELM-MAS, ELM [3] and SVM [51]

Algorithm	Abalone	Strike	Space-ga
	RMSE	RMSE	RMSE
ELM-MAS	**0.0174**	**0.1637**	**0.0068**
SVM [51]	0.0764	0.2282	0.0648
ELM [3]	0.0761	0.2985	0.0624

V.REAL WORLD APPLICATION: NO_X EMISSION OF POWER GENERATION PLANT

In the context of real world application in this study, the NO_x emission of an open cycle gas turbine in a power generation plant (located at Port Dickson, Malaysia) had been investigated [52]. The objective is to establish a neural network model for the prediction of NO_x emission. There are 150 input attributes taken from the parameters of the power generation plant such as the loading of the gas turbine, temperature, pressure and etc. The targeted output is the quantity of NO_x (in ppm) emitted from the gas turbine.

A total of 3,405 data samples had been collected for training and testing of ELM-MAS. Out of 3,405 data samples, 2,270 are used for training while left behind 1135 are used for testing as shown in Table 4. An experiment is conducted on the testing datasets for fifty runs and the average results are recorded. A summary result for No_x emission with different activation functions is shown in Table 5. The best RMSE result in Table 5 is 0.0049 using Sigmoid activation function.

Table. 4 Specification of No_x Emission Datasets [52]

Dataset	# Attributes	# Training	# Testing
No_x emission	150	2,270	1,135

Table. 5 Summary of ELM-MAS for No_x Emission Datasets with Different Activation Functions

Activation Functions	No_x Emission	
	RMSE	# Hidden Neurons
RBF	0.0061	36
Sigmoid	**0.0049**	**9**
Gaussian	0.0073	29
LaplaceAct	0.0064	17
Laplace Basis	0.0058	30

In the experiment of using ELM, two-thirds of the data samples are used for training while the remaining one-third are used for the validation process to decide the most appropriate number of neurons of the parent ELM (i.e.*L*). The validation and training processes for sigmoid activation function (SigAct) of ELM are started by setting $L = 50$ units and then increased by an increment of 50 units. An experiment is conducted on the testing datasets for fifty runs and the average results are recorded. As an example, Table 6 showed the testing processes based on SigAct. Based on the results of RMSE (Table 6), the best RMSE obtained is 0.027086. The max and min of RMSE for $L = 200$ are 0.028416 and 0.026952 respectively. Using the result in Table 6 to compare with Table 5, the RMSE results of ELM is higher than RMSE of ELM-MAS due to the complexity of hidden neurons in ELM.

Table. 6 RMSE of NO_x emission for ELM

L	RMSE
50	0.036684
100	0.029197
150	0.027448
200	**0.027086**
250	0.027091
300	0.027233

VI.CONCLUSIONS

In essence, a new two layers of ELMs model called ELM-MAS is established for regression. The ELM-MAS model is certified by consuming benchmark datasets for instance abalone, strike and space-ga. The experimental outcomes demonstrates that the proposed model is better preferable than SVM[51] and ELM [3], as shown in Table 3. Moreover, the ELM-MAS model is assessed by applying it on the real world application which is No_x emission of power generation plant. Thus far, our results demonstrated that the RMSE of ELM-MAS for No_x emission is better that ELM.

Although results obtained from the abalone, strike and space-ga as well as No_x emission are reassuring, advance study with datasets from diverse application fields is essential for validating the capability of ELM-MAS application in real world.

ACKNOWLEDGMENTS

This work was supported by Universiti Tenaga National under grant J510050684.

REFERENCES

1. Huang, G.-B. and L. Chen, Convex incremental extreme learning machine. Neurocomputing, 2007. 70(16-18): p. 3056-3062.
2. Huang, G.-B. and L. Chen, Enhanced random search based incremental extreme learning machine. Neurocomputing, 2008. 71(16-18): p. 3460-3468.
3. Huang, G.-B., et al., Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012. 42(2): p. 513-529.
4. Huang, G.-B., et al., Can threshold networks be trained directly? IEEE Transactions on Circuits and Systems II: Express Briefs, 2006. 53(3): p. 187-191.
5. Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks. Neural networks, 2004. 2: p. 985-990.
6. Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: theory and applications. Neurocomputing, 2006. 70(1-3): p. 489-501.
7. Yap, K.S. and H.J. Yap, Daily maximum load forecasting of consecutive national holidays using OSELM-based multi-agents system with weighted average strategy. Neurocomputing, 2012. 81: p. 108-112.
8. Huang, G., et al., Trends in extreme learning machines: A review. Neural Networks, 2015. 61: p. 32-48.
9. Liu, X., et al., Is extreme learning machine feasible? A theoretical assessment (Part I). IEEE Transactions on Neural Networks and Learning Systems, 2015. 26(1): p. 7-20.
10. Lin, S., et al., Is extreme learning machine feasible? A theoretical assessment (Part II). IEEE Transactions on Neural Networks and Learning Systems, 2015. 26(1): p. 21-34.

11. Huang, G.-B., X. Ding, and H. Zhou, Optimization method based extreme learning machine for classification. Neurocomputing, 2010. 74(1-3): p. 155-163.
12. Song, Y., J. Crowcroft, and J. Zhang, Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine. Journal of neuroscience methods, 2012. 210(2): p. 132-146.
13. Zhang, Y. and P. Zhang, Optimization of nonlinear process based on sequential extreme learning machine. Chemical engineering science, 2011. 66(20): p. 4702-4710.
14. Zhao, G., et al. Enhanced extreme learning machine with stacked generalization. in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). 2008. IEEE.
15. Minhas, R., A.A. Mohammed, and Q.J. Wu, Incremental learning in human action recognition based on snippets. IEEE Transactions on Circuits and Systems for Video Technology, 2012. 22(11): p. 1529-1541.
16. Hasan, M. and A.K. Roy-Chowdhury, Incremental learning of human activity models from videos. Computer Vision and Image Understanding, 2016. 144: p. 24-35.
17. Zhou, Y., J. Peng, and C.P. Chen, Extreme learning machine with composite kernels for hyperspectral image classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2015. 8(6): p. 2351-2360.
18. Yang, Y., Y. Wang, and X. Yuan, Bidirectional extreme learning machine for regression problem and its learning effectiveness. IEEE Transactions on Neural Networks and Learning Systems, 2012. 23(9): p. 1498-1505.
19. Nizar, A., Z. Dong, and Y. Wang, Power utility nontechnical loss analysis with extreme learning machine method. IEEE Transactions on Power Systems, 2008. 23(3): p. 946-955.
20. Yan, Z. and J. Wang, Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. IEEE transactions on neural networks and learning systems, 2014. 25(3): p. 457-469.
21. Van Heeswijk, M., et al. Adaptive ensemble models of extreme learning machines for time series prediction. in International Conference on Artificial Neural Networks. 2009. Springer.
22. Lan, Y., Y.C. Soh, and G.-B. Huang, Ensemble of online sequential extreme learning machine. Neurocomputing, 2009. 72(13-15): p. 3391-3395.
23. Van Heeswijk, M., et al., GPU-accelerated and parallelized ELM ensembles for large-scale regression. Neurocomputing, 2011. 74(16): p. 2430-2437.
24. Quteishat, A., et al., A neural network-based multi-agent classifier system. Neurocomputing, 2009. 72(7-9): p. 1639-1647.
25. Sun, Z.-L., et al., Sales forecasting using extreme learning machine with applications in fashion retailing. Decision Support Systems, 2008. 46(1): p. 411-419.
26. Gwebu, K., J. Wang, and M.D. Troutt, Constructing a multi-agent system: an architecture for a virtual marketplace. Intelligent Decision Support Systems in Agent-Mediated Environments. IOS Press, Amsterdam, 2005.
27. Das, S., et al., AI doctor: an intelligent approach for medical diagnosis, in Industry Interactive Innovations in Science, Engineering and Technology. 2018, Springer. p. 173-183.
28. Salem, H., G. Attiya, and N. El-Fishawy, A survey of multi-agent based intelligent decision support system for medical classification problems. International Journal of Computer Applications, 2015. 123(10).
29. Hudson, D. and M. Cohen. Use of intelligent agents in the diagnosis of cardiac disorders. in Computers in Cardiology. 2002. IEEE.
30. Oprea, M., ABVE-Frame: An agent-based virtual enterprise development framework. AI Communications, 2017. 30(2): p. 117-140.
31. Borrajo, M.L. and J.M. Corchado. An Agent-Based Virtual Organization for Risk Control in Large Enterprises. in International Conference on Knowledge Management in Organizations. 2018. Springer.
32. Nyongesa, H.O., G.W. Musumba, and N. Chileshe, Partner selection and performance evaluation framework for a construction-related virtual enterprise: a multi-agent systems approach. Architectural Engineering and Design Management, 2017. 13(5): p. 344-364.
33. Tolk, A., An agent-based decision support system architecture for the military domain. 2005.
34. Naseem, A., et al., Decision support system for optimum decision making process in threat evaluation and weapon assignment: Current status, challenges and future directions. Annual reviews in control, 2017. 43: p. 169-187.
35. Sampaio, R.F., et al., Automatic restoration system for power distribution networks based on multi-agent systems. IET Generation, Transmission & Distribution, 2017. 11(2): p. 475-484.
36. Liu, M., et al., Intelligent assembly system for mechanical products and key technology based on internet of things. Journal of Intelligent Manufacturing, 2017. 28(2): p. 271-299.
37. Kortenkamp, D., R. Simmons, and D. Brugali, Robotic systems architectures and programming, in Springer Handbook of Robotics. 2016, Springer. p. 283-306.
38. Singh, R., A. Salam, and L. Iyer, Using agents and XML for knowledge representation and exchange: an intelligent distributed decision support architecture (IDDSA). AMCIS 2003 Proceedings, 2003: p. 239.
39. Ossowski, S., et al. Designing multiagent decision support system—The case of transportation management. in null. 2004. IEEE.
40. Karavas, C.-S., et al., A multi-agent decentralized energy management system based on distributed intelligence for the design and control of autonomous polygenerationmicrogrids. Energy Conversion and Management, 2015. 103: p. 166-179.
41. Singh, B. and A. Gupta, Recent trends in intelligent transportation systems: a review. Journal of Transport Literature, 2015. 9(2): p. 30-34.
42. Le Pira, M., et al., Towards a decision-support procedure to foster stakeholder involvement and acceptability of urban freight transport policies. European Transport Research Review, 2017. 9(4): p. 54.
43. Ossowski, S., et al. Engineering agent systems for decision support. in International Workshop on Engineering Societies in the Agents World. 2002. Springer.
44. Trappey, A.J., et al., Intelligent engineering asset management system for power transformer maintenance decision supports under various operating conditions. Computers & Industrial Engineering, 2015. 84: p. 3-11.
45. Ghadimi, P., F.G. Toosi, and C. Heavey, A multi-agent systems approach for sustainable supplier selection and order allocation in a partnership supply chain. European Journal of Operational Research, 2018. 269(1): p. 286-301.
46. Lopez-Lorca, A.A., et al., Supporting agent oriented requirement analysis with ontologies. International Journal of Human-Computer Studies, 2016. 87: p. 20-37.
47. Jenkins, D.P., M. BOYD, and C. LANGLEY. Using the decision ladder to reach a better design. in The Ergonomics Society Annual Conference. 19e21 April. 2016.
48. Marom, N.D., L. Rokach, and A. Shmilovici. Using the confusion matrix for improving ensemble classifiers. in 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel. 2010. IEEE.
49. Liao, S. and C. Feng, Meta-ELM: ELM with ELM hidden nodes. Neurocomputing, 2014. 128: p. 81-87.
50. Guo, Y., et al. Meta-learning for parallel data mining. in Proceedings of the seventh parallel computing workshop. 1997.
51. Liang, N.-Y., et al., A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Transactions on neural networks, 2006. 17(6): p. 1411-1423.
52. SaifulIdzwan, B., C.C. Phing, and T.S. Kiong, Prediction of Nox using support vector machine for gas turbine emission at Putrajaya power station. J Adv SciEng Res, 2014. 4: p. 37-46.

AUTHORS PROFILE

Chong Tak Yaw, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.

Keem Siah Yap, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.

ShenYuong Wong, Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Malaysia

Chin Hooi Tan, Business Innovation Incubator, Tenaga Nasional Berhad, Malaysia